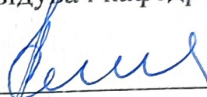


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Львівський національний університет імені Івана Франка
Факультет прикладної математики та інформатики
Кафедра математичного моделювання соціально-економічних процесів

Оновлено та затверджено
на засіданні
кафедри математичного моделювання
соціально-економічних процесів
факультету прикладної математики та
інформатики
Львівського національного університету
імені Івана Франка
(протокол № 2 від 27.08.2025 р.)

Завідувач кафедри

 Петро СЕНЬО

Силабус з навчальної дисципліни
“Прикладне програмування на Python”,
що викладається в межах ОПП “Системний аналіз і управління.
Інтелектуальний аналіз даних”
першого (бакалаврського) рівня вищої освіти для здобувачів з
спеціальності 124 – системний аналіз

Львів 2025 р.

Назва дисципліни	Прикладне програмування на Python
Адреса викладання дисципліни	Головний корпус ЛНУ ім. І. Франка м. Львів, вул. Університетська 1
Факультет та кафедра, за якою закріплена дисципліна	Факультет прикладної математики та інформатики Кафедра математичного моделювання соціально-економічних процесів
Галузь знань, шифр та назва спеціальності	12 – інформаційні технології 124 – системний аналіз
Викладачі дисципліни	Лисецька Олександра Юріївна, доктор філософії за спеціальністю «Математика», доцент кафедри математичного моделювання соціально-економічних процесів Коркуна Андрій Михайлович, старший викладач кафедри математичного моделювання соціально-економічних процесів
Контактна інформація викладачів	oleksandra.lysetska@lnu.edu.ua ; https://ami.lnu.edu.ua/employee/lysetska-o-yu ; Головний корпус ЛНУ ім. І. Франка, каб. 361. м. Львів, вул. Університетська, 1 andriy.korkuna@lnu.edu.ua https://ami.lnu.edu.ua/employee/korkuna-andrij ; Головний корпус ЛНУ ім. І. Франка, каб. 361. м. Львів, вул. Університетська, 1
Консультації з питань навчання по дисципліні відбуваються	Консультації в день проведення лекцій/лабораторних занять (за попередньою домовленістю). Можливі онлайн консультації через Zoom чи Microsoft Teams. Для погодження часу онлайн консультацій потрібно писати на електронну пошту викладача.
Сторінка курсу	https://ami.lnu.edu.ua/course/prykladne-prohramuvannia-na-python https://e-learning.lnu.edu.ua/course/view.php?id=6641
Інформація про дисципліну	Дисципліна «Прикладне програмування на Python» є вибірковою дисципліною з спеціальності 124 – системний аналіз для ОПП «Системний аналіз і управління. Інтелектуальний аналіз даних», яка викладається у восьмому семестрі в обсязі 5 кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Коротка анотація дисципліни	Вивчення спеціалізованих бібліотек і засобів мови Python, необхідних для оволодіння прийомами прикладного програмування, орієнтованими на застосування за фахом в майбутній роботі. Формування системи знань про бібліотеки, методи, алгоритми і прийоми програмування типових прикладних задач в рамках розробки програмних проєктів. Формування практичних навичок складання фахових прикладних програм, структур даних, моделей за розділами, викладеними в переліку тем курсу. Курс використовує низку знань і навиків, які були отримані студентами в інших раніше викладених курсах за розділами програмування і суміжних з ними.

<p>Мета та цілі дисципліни</p>	<p><i>Метою</i> вивчення нормативної дисципліни “Прикладне програмування на Python” є вивчення спеціалізованих бібліотек і засобів мови Python, пов'язаних з прикладним програмуванням типових частин реальних проєктів; розуміння предмета дисципліни, зв'язку з іншими суміжними засобами і методами програмування, необхідними для цілей майбутньої розробки програмних проєктів; вміння застосовувати алгоритмічну мову Python в обсязі задач розділів курсу; вивчення сучасних методів і алгоритмів програмування прикладних задач; отримання навиків розробки процедур проектування, програмування, тестування і налагодження типових прикладних програм мовою Python; вміння використовувати стандартні і тематичні бібліотеки Python, середовище програмування Python.</p> <p><i>Цілями</i> вивчення навчальної дисципліни є сформувати у студентів теоретичні знання необхідні для алгоритмічного розв'язування задач прикладного програмування та сформувати необхідні практичні навички роботи з різноманітними бібліотеками та модулями мови програмування Python, що дозволяють розробити програмну реалізацію цих прикладних задач.</p>
<p>Література для вивчення дисципліни</p>	<p style="text-align: center;">Основна література</p> <ol style="list-style-type: none"> 1. Mather B. Time series with Python. – Abiproduct., 2020. – 185 p. 2. Klein B. Applied Python. – Bodensee., 2021. – 2021. – 111p. 3. Мова Python для інженерних та наукових задач [Електронний ресурс] : підруч. / О. Ю. Горобець, С. В. Горобець, К. Ю. Хахно ; КПІ ім. Ігоря Сікорського. — Київ : КПІ, 2024. — 277 с. — Електронні текстові дані. — Режим доступу: https://ela.kpi.ua/items/29c3b487-d8b2-4fb1-b5b6-2c2332efa518 4. Python Software Foundation. The Python Tutorial [Електронний ресурс]. – Режим доступу: https://docs.python.org/3/tutorial/index.html 5. Python Software Foundation. Python 3.7.12 documentation [Електронний ресурс]. – Режим доступу: https://docs.python.org/3.7/ 6. Лисецька О. Ю., Фундак Л. І., Ярошко С. М. Бібліотека NumPy: теоретичні основи і застосування : навч. посіб. — Львів, 2025. — 180 с. 7. Олещенко Л.М. Технології оброблення великих даних: комп'ютерний практикум з дисципліни «Технології оброблення великих даних». – КПІ ім. Ігоря Сікорського, 2021. – 85 с. 8. Перелік файлових форматів. Матеріал з Вікіпедії — вільної енциклопедії. [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Перелік_файлових_форматів 9. Список структур даних. Матеріал з Вікіпедії — вільної енциклопедії. [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Список_структур_даних <p style="text-align: center;">Додаткова література</p> <ol style="list-style-type: none"> 10. Chun Wesley J. Core Python Application Programming. Third Edition. – Pearson Education, Inc., 2012. 11. Ignatenko O. Game theory in Peaky Blinders. Episode 1. [Electronic Resource]. – Mode of access to the resource: https://medium.com/@oleksii_ignatenko/game-theory-in-peaky-blinders-episode-1-db8532dfd577 12. Lutz M. Programming Python, Forth Edition. – O'Reilly Media Inc., 2011.

	<p>13. Lutz M. Learning Python, 5th Edition. – O'Reilly Media Inc., 2013. – 1648 p.</p> <p>14. Microsoft Ignite. Visual Studio documentation [Electronic Resource]. – Mode of access to the resource: https://docs.microsoft.com/enus/visualstudio/windows/?f1url=%3FappId%3DDev15IDEF1%261%3DenUS%26k%3Dk(MSDNSTART)%26rd%3Dtrue&view=vs-2019</p> <p>15. Vaughan L. Real-World Python: A Hacker's Guide to Solving Problems with Code. – No Starch Press., 2020. – 360p.</p> <p>16. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест, Кліфорд Стайн. Вступ до алгоритмів. — К. : К. І. С., 2019. — 1288 с. ISBN 978-617684-239-2 [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Вступ_до_алгоритмів#Український_переклад</p>
Обсяг курсу	Загальний обсяг: 150 годин. Аудиторних занять: 56 год., з них 28 год. лекцій та 28 години лабораторних робіт. Самостійної роботи: 94 год.
Очікувані результати навчання	<p>Після завершення цього курсу студент буде :</p> <p>Знати:</p> <ul style="list-style-type: none"> – сучасні методи і алгоритми програмування прикладних задач мовою Python; – засоби спеціалізованих бібліотек мови Python, пов'язані з прикладним програмуванням типових частин реальних проєктів; – процедури моделювання і застосування структур даних мовою Python, необхідних для реальних проєктів; – методи організації виконання сценаріїв (Python-програм); – технології роботи з internet-файлами, формати файлів; – технології роботи з часовими рядами та великими даними; <p>Вміти:</p> <ul style="list-style-type: none"> – застосовувати засоби мови Python і спеціалізованих бібліотек для програмування типових прикладних задач; – складати, тестувати і налагоджувати прикладні функції частин реальних проєктів мовою Python; – реалізувати вимоги замовників до розробки програмних проєктів мовою Python; – аналізувати і модифікувати прикладні програми до зміни вимог; – розв'язувати задачі інтелектуального аналізу даних за допомогою бібліотек Python. <p>Курс забезпечує набуття таких компетентностей та програмних результатів навчання:</p> <p>Загальні компетентності:</p> <p>Здатність до абстрактного мислення, аналізу та синтезу. Здатність до пошуку, оброблення та аналізу інформації з різних джерел. Здатність спілкуватися з представниками інших професійних груп різного рівня (з експертами з інших галузей знань/видів економічної діяльності). Здатність розробляти проєкти та управляти ними.</p> <p>Спеціальні компетентності:</p> <p>Здатність інтегрувати знання та здійснювати системні дослідження, застосовувати методи математичного та інформаційного моделювання складних систем та процесів різної природи.</p>

	<p>Здатність розробляти системи підтримки прийняття рішень та рекомендаційні системи.</p> <p>Здатність оцінювати ризики, розробляти алгоритми управління ризиками в складних системах різної природи.</p> <p>Програмні результати навчання:</p> <p>Будувати та досліджувати моделі складних систем і процесів застосовуючи методи системного аналізу, математичного, комп'ютерного та інформаційного моделювання.</p> <p>Застосовувати методи машинного навчання та інтелектуального аналізу даних, математичний апарат нечіткої логіки, теорії ігор та розподіленого штучного інтелекту для розв'язання складних задач системного аналізу.</p>
Ключові слова	Python, синтаксис, семантика, оператор, сценарій, структури даних, словник, стек, черга, файл, каталог, потік, процес, інтернет, сервер, вебсторінка, формат xml, формат json, python+json, програмний проєкт, архітектура програмного проєкта, парадигма програмування, технологія програмування.
Формат курсу	Очний.
Теми	Теми подані у таблиці Схемі курсу «Прикладне програмування на Python» нижче.
Підсумковий контроль, форма	Залік.
Пререквізити	<p>Для вивчення курсу студенти потребують базових знань з таких дисциплін, як:</p> <ul style="list-style-type: none"> - Програмування; - Теорія ймовірностей та математична статистика; - Дискретна математика; - Математична логіка; - Основи інформаційних технологій.
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	<p>Лекції з мультимедійними презентаціями та з демонстрацією прийомів прикладного програмування мовою Python;</p> <p>лабораторні заняття у вигляді проєктування алгоритмів і програм, складання моделей реальних проєктів, виконання практичних завдань та проєктів;</p> <p>самостійне опрацювання навчальних матеріалів: підручників, конспектів лекцій, електронних ресурсів, готових програм, додаткових матеріалів, розміщених у хмарному сховищі (Microsoft Teams, Google Диск).</p> <p>Обговорення теоретичного та практичного матеріалу на онлайн сервісах, формулювання творчих завдань для студентів, виконання яких готує до вивчення нового теоретичного і практичного матеріалу.</p>
Необхідне обладнання	Комп'ютер із програмним забезпеченням Visual Studio, Anaconda Navigator (Jupyter Notebook) або PyCharm та встановленими бібліотеками numpy, pandas, matplotlib, scipy, nashpy, seaborn, plotly, pytorch, TypeVar, Generic. Internet доступ до обчислювального кластера, проєктор, обладнання спеціалізованої лабораторії математичного та комп'ютерного

	<p>моделювання (офіс 365). Для організації відеоконференцій використовуються програми Zoom (безкоштовна ліцензія Zoom Meetings Education) або MS Teams (ліцензія Microsoft 365 A5)</p>				
<p>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</p>	<p>Оцінювання проводиться за 100-бальною шкалою.</p>				
	<p>Оцінка за шкалою ECTS</p>	<p>Оцінка в балах</p>	<p>Оцінка за національною шкалою</p>		
			<p>Екзамен, диференційований залік</p>		<p>Залік</p>
	A	90-100	відмінно	5	<p>Зараховано</p>
	B	81-89	добре	4	
	C	71-80			
	D	61-70			
	E	51-60	задовільно	3	<p>не зараховано</p>
	F	21-50	незадовільно	2	
	FX	0-20	незадовільно (без права перездачі)	2	
<p>Поточне оцінювання: впродовж семестру студент може отримати максимум 100 балів. З них:</p> <p>за роботу на лабораторних заняттях: максимальна кількість – 42 бали (по 3 бали за кожну лабораторну роботу).</p> <p>за виконання індивідуальних завдань: максимальна кількість балів – 34:</p> <ul style="list-style-type: none"> – за перше індивідуальне завдання – 8 балів, – за друге індивідуальне завдання – 8 балів, – за третє індивідуальне завдання – 8 балів, – за четверте індивідуальне завдання – 10 балів; <p>колоквіум: максимальна кількість – 14 балів (13 тестових завдань по 1б. та 1 практичне завдання – 1б.).</p> <p>комплексне контрольне опитування: максимальна кількість балів – 10 (усне опитування, що складається з 3 теоретичних питань різної складності, 1 та 2 питання оцінюються максимум у 3 бали, а 3 питання – максимум 4 бали).</p>					
<p align="center">Критерії оцінювання лабораторних робіт.</p>					
<p>№ 1-7 (5 балів)</p>	<p align="center">Критерії оцінювання</p>				
<p>5 балів</p>	<p>студент повністю виконав умови завдання, алгоритм реалізовано правильно, відповідає на всі запитання, пов'язані з тематикою завдання, проводить чіткий аналіз та порівняння отриманих результатів, пропонує інші підходи до вирішення поставленого завдання;</p>				
<p>4 балів</p>	<p>студент повністю виконав умови завдання, на деякі запитання, алгоритм реалізовано правильно, пов'язані з тематикою завдання, відповідає з незначними неточностями, проводить аналіз отриманих результатів з незначними неточностями;</p>				
<p>3 бали</p>	<p>студент виконав завдання з незначними помилками, але самостійно їх виправляє, якщо на них вкаже викладач, на деякі запитання, пов'язані з тематикою завдання, відповідає з неточностями, проводить аналіз отриманих результатів з неточностями;</p>				
<p>2 бали</p>	<p>студент виконав завдання частково, алгоритм реалізовано з помилками, які самостійно не може виправити, переважно не відповідає на запитання;</p>				
<p>1 бал</p>	<p>студент виконав завдання частково або з грубими помилками, які</p>				

	самостійно не може виправити, демонструє незнання матеріалу;
0 балів	студент не виконав завдання.
Критерії оцінювання питань №1 та №2 (комплексне контрольне опитування):	
3 бали	– студент(-ка) повністю розкрив(-ла) тему, схема доповіді логічна, впевнено відповідає на всі запитання, які пов'язані з тематикою доповіді та демонструє глибокі знання;
2 бали	– студент(-ка) повністю розкрив(-ла) тему доповіді з незначними неточностями, схема доповіді логічна, відповідає на більшість запитань, які пов'язані з тематикою доповіді;
1 бал	– студент(-ка) в основному розкрив(-ла) тему доповіді з незначними неточностями, демонструє слабкі знання, відповідає лише на окремі запитання, які пов'язані з тематикою доповіді;
0 балів	– студент(-ка) не підготував(-ла) доповіді або тему доповіді не розкрито і під час захисту студент не може відповісти на жодне запитання за тематикою доповіді.
Критерії оцінювання питання №3 (комплексне контрольне опитування):	
4 балів	– студент(-ка) повністю розкрив(-ла) тему доповіді, схема доповіді логічна, впевнено відповідає на всі запитання, які пов'язані з тематикою доповіді та демонструє глибокі знання;
3 бали	– студент(-ка) повністю розкрив тему доповіді з незначними неточностями, схема доповіді логічна, відповідає на більшість запитань, які пов'язані з тематикою доповіді;
1-2 бали	– студент(-ка) в основному розкрив(-ла) тему доповіді з незначними неточностями, демонструє слабкі знання, відповідає лише на окремі запитання, які пов'язані з тематикою доповіді;
0 балів	– студент(-ка) не підготував(ла) доповіді або тему доповіді не розкрито і під час захисту студент не може відповісти на жодне запитання за тематикою доповіді.
<p>Кожен(-на) студент(-ка) отримує окреме (персональне) завдання, для розв'язування якого потрібно використати можливості одного чи декількох із розглянутих на лекціях пакетів чи модулів. Кожне індивідуальне завдання відповідає блоку питань, відображених у Схемі курсу. Для поставленої задачі необхідно написати скрипт (програму) на мові Python, дотримуючись інструкцій вказаних в умові, відлагодити програми і отримати результати тестових прикладів.</p>	
Критерії оцінювання індивідуальних завдань №1-3 (4):	
Максимальна кількість балів	Критерій оцінювання
Для кожного завдання 8 (10) балів	
8 балів (10-9 балів)	Студент(-ка) повністю виконав(-ла) умови завдання з використанням лише тих відповідних комп'ютерних пакетів та бібліотек Python, які

		пов'язані із тематикою завдання алгоритми реалізовано правильно, відповідає на всі запитання, пов'язані з тематикою завдання, проводить чіткий аналіз та порівняння отриманих результатів.
	7 балів (8-7 балів)	Студент(-ка) повністю виконав(-ла) умови завдання без використання відповідних комп'ютерних пакетів, алгоритми, які пов'язані з тематикою завдання, реалізовано правильно, на деякі запитання, відповідає з незначними неточностями, проводить аналіз отриманих результатів з незначними неточностями.
	6 балів (6-5 балів)	Студент(-ка) виконав(-ла) завдання лише з використанням відповідних комп'ютерних пакетів з незначними помилками, але самостійно їх виправляє, якщо на них вкаже викладач, на деякі запитання, пов'язані з тематикою завдання, відповідає з неточностями, проводить аналіз отриманих результатів з неточностями.
	4 – 3 бали	Студент(-ка) виконав(-ла) завдання частково і лише з використанням відповідних комп'ютерних пакетів, алгоритм реалізовано з помилками, які частково може виправити, якщо на них вкаже викладач, на запитання відповідає з помилками, проводить аналіз отриманих результатів з помилками.
	2 – 1 бали	Студент(-ка) виконав(-ла) завдання частково, або з грубими помилками, які самостійно не може виправити, переважно не відповідає на запитання.
	0 балів	Студент(-ка) не виконав(-ла) завдання.
<p>Для кожного завдання встановлено терміни здачі. Роботи, які здаються із порушенням термінів без поважних причин, оцінюються меншою кількістю балів. Так, запізнення до 7 днів зменшує максимальну кількість балів на 25%, від 8 до 14 днів – на 50 %, від 15 до 21 днів – на 75 %, більше 21 днів – на 90%.</p> <p>Письмові роботи: Очікується, що студенти виконають одну письмову роботу (колоквіум) і подадуть письмові (електронні) звіти про виконання чотирьох індивідуальних завдань.</p> <p>Усунення академічної заборгованості. Академічна заборгованість виникає за сумарного балу менше 51. Дозволяється один курс перездавати не більше двох раз: Форма 2 та Форма К (приймає комісія з 3 викладачів кафедри на чолі із завідувачем кафедри). На кожному етапі студент стартує з 0 балів. Прохідний бал – не менше 51. Якщо на Формі К набрано менше 51 балу – дозволяється повторне вивчення курсу.</p> <p>Форма 2 Студент зобов'язаний виконати :</p> <p>1. Усі нездані або низькооцінені лабораторні роботи (визначаються</p>		

викладачами курсу на основі поточного оцінювання студента, мінімум 8 робіт, що охоплюють ключові теми: базові типи та структури даних, регулярні вирази, робота з файлами/каталогами, API/requests/plotly, теорія ігор (nashpy), теорія графів (NetworkX), JSON/XML, бази даних, наукові обчислення (numpy/pandas/scipy), аналіз пропусків). Кожна лабораторна робота оцінюється максимально **5 балів**.

2. Усі 4 індивідуальні завдання (навіть якщо деякі були здані раніше – виконуються заново або переоформлюються з урахуванням зауважень).

○ Завдання №1-3: максимально **8 балів** кожне.

○ Завдання №4: максимально **10 балів**.

3. Повний звіт-проект, що інтегрує результати всіх 4 індивідуальних завдань та демонструє архітектуру програмного проекту (аналіз результатів, тестування, документація) — максимально **15 балів**.

Максимальна кількість балів: 40 (лабораторні) + 34 (індивідуальні) + 15 (звіт) + 11 (додаткові бали за усний захист/відповіді на питання викладача) у сумі 100 балів.

Форма К (приймає комісія)

Студент виконує:

1. Вибірку з 10 практичних завдань, визначену комісією (по 1-2 з кожного ключового блоку курсу: базові структури, регулярні вирази, файли/каталоги, API/plotly, nashpy, NetworkX, JSON/XML, бази даних, наукові обчислення, пропуски в даних). Кожне завдання оцінюється максимально **6 балів**.

2. Одне комплексне індивідуальне завдання-проект (нове, інтеграційне, наприклад: отримання даних з API, обробка JSON/XML, аналіз з пропусками, візуалізація/plotly/NetworkX, знаходження рівноваги Неша) – максимально **20 балів**.

3. Повний звіт за всім курсом (включає описи всіх тем, код, результати, аналіз та документацію проекту, навіть якщо окремі частини були здані раніше) – максимально **20 балів**.

Максимальна кількість балів: 60 (10 завдань) + 20 (комплексне завдання) + 20 (звіт) у сумі 100 балів.

Критерії оцінювання завдань (застосовуються на обох формах)

Для лабораторних завдань (максимум 5–6 балів):

	Критерії оцінювання
5-6 балів	студент повністю виконав завдання, код правильний та оптимальний, використані лише відповідні бібліотеки курсу, студент чітко аналізує результати, пропонує альтернативні підходи, впевнено відповідає на всі теоретичні та практичні питання.;
4 балів	студент повністю виконав умови завдання, на деякі запитання, алгоритм реалізовано правильно, пов'язані з тематикою завдання, відповідає з незначними неточностями, проводить аналіз отриманих результатів з незначними неточностями;
3 бали	студент виконав завдання з незначними помилками, але самостійно їх виправляє, якщо на них вкаже викладач, на деякі запитання, пов'язані з тематикою завдання, відповідає з неточностями, проводить аналіз отриманих результатів з неточностями;

2 бали	студент виконав завдання частково, алгоритм реалізовано з помилками, які самостійно не може виправити, переважно не відповідає на запитання;
1 бал	студент виконав завдання частково або з грубими помилками, які самостійно не може виправити, демонструє незнання матеріалу;
0 балів	студент не виконав завдання.

Для індивідуальних та комплексних завдань (8–20 балів):

Оцінювання за чотирма критеріями (пропорційно розподілені бали):

1. Правильність та повнота реалізації (використання відповідних бібліотек, алгоритмів, тестування) — 40–50% балів.
2. Якість коду (читабельність, коментарі, структура, дотримання стандартів Python) — 20%.
3. Аналіз результатів (порівняння, обґрунтування вибору методів, висновки) — 20–30%.
4. Усний захист/відповіді на питання (пояснення архітектури, відповіді на теоретичні питання за темою) — 10–20%.

Для звіту (15–20 балів):

- 80–100% балів: звіт повний, логічно структурований, містить код, результати, аналіз, документацію, відсутні помилки.
- 60–79%: звіт повний, але з незначними неточностями в аналізі чи оформленні.
- 40–59%: звіт частковий, аналіз поверховий.
- менше 40%: звіт неповний або з грубими помилками.

Академічна доброчесність. Очікується, що роботи студентів будуть їх оригінальними дослідженнями чи міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів становлять, але не обмежують, приклади можливої академічної недоброчесності. Використання моделей генеративного штучного інтелекту повинно бути аргументованим із чітко зазначеним переліком, для виконання яких завдань було застосовано ШІ. Виявлення ознак академічної недоброчесності в письмовій роботі студента є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману. Жодні форми порушення академічної доброчесності не толеруються.

Відвідання занять є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції та практичні заняття курсу. Студенти повинні інформувати викладача про неможливість відвідати заняття. У будь-якому випадку студенти зобов'язані дотримуватися термінів визначених для виконання всіх видів письмових робіт та індивідуальних завдань, передбачених курсом.

Література. Уся література, яку студенти не зможуть знайти самостійно, буде надана викладачем виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих.

Політика виставлення балів. Враховуються бали набрані при поточному оцінюванні на практичних заняттях, колоквиумі, комплексному контрольному оцінюванні та бали за індивідуальні роботи. При цьому обов'язково враховуються присутність на заняттях та активність студента

	під час практичного заняття; недопустимість пропусків та запізнь на заняття; користування мобільним телефоном, планшетом чи іншими мобільними пристроями під час заняття в цілях не пов'язаних з навчанням; списування та плагіат; несвоєчасне виконання поставленого завдання і т.д.
Питання	<ol style="list-style-type: none"> 1. Середовища розробки python-програм. Базове середовище IDLE. 2. Параметри середовищ для розробки програм. Взаємодія з операційною системою. 3. Базові типи даних мови Python. Принципи будови системи даних. Змінні величини, об'єкти, посилання. 4. Поліморфізм операцій. Типові бібліотеки функцій. 5. Алгоритми програмування задач типового змісту. Використання типів Python і засобів бібліотек (TypeVar, Generic). Вбудовані структури даних мови Python. 6. Словники. Базові операції над словниками. Типові задачі з використанням словників. 7. Системи кодування літер. Використання бібліотек TypeVar, Generic. 8. Основні принципи роботи з регулярними виразами за допомогою бібліотеки re. 9. Режими відкривання файла. Сканери файлів. Модулі та інструменти для роботи з файлами і каталогами. 10. Задачі пошуку файлів за різними критеріями. Відображення характеристик файла. 11. Задачі операцій з файлами і каталогами. Інструменти розробника. 12. Основні принципи роботи з модулем request. API. 13. Матрична гра. Рівновага Неша. Пакет nashpy. 14. Основні поняття теорії графів. Бібліотека NetworkX. Моделювання соціально-економічних процесів засобами Python. 15. Текстові формати обміну даними Json та XML. Сервери і ресурси форматів Json та XML. Перетворення json(xml)-файлів в python-структури. 16. Отримання json(xml)-даних. Дослідження структури json(xml)-файла. Приклади задач опрацювання даних. 17. Бази даних. Робота з базами даних в python. Бібліотеки для роботи з реляційними та нереляційними базами даних. 18. Наукові обчислення за допомогою python. 19. Механізми породження пропусків. Методи заповнення пропусків. Заповнення середніми. Заповнення з підбором. Заповнення за регресією 20. Тестування та відлагодження програм.
Опитування	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенні курсу.

Схема курсу «Прикладне програмування на Python»

Тиж.	Тема, план, короткі тези	Форма діяльності (заняття)	Література. Ресурси в інтернеті	Завдання, год.	Термін виконання
1	Тема 1. Вступ до прикладного програмування на Python (Середовища розробки Python-програм. Базове середовище IDLE. Параметри середовищ для розробки програм. Взаємодія з операційною системою).	Лекція, самостійна робота	[4, 9, 12-14, 16]	2, 2	1 тиждень
	Тема 1. Вступ до прикладного програмування на Python (Базові типи даних мови Python. Принципи будови системи даних. Змінні величини, об'єкти, посилання. Поліморфізм операцій. Типові бібліотеки функцій.).	Лабораторна, самостійна робота	[4, 5, 9, 12-14, 16]	2, 3	Під час заняття
2	Тема 2. Робота з різними типами даних (Алгоритми програмування задач типового змісту. Використання типів Python і засобів бібліотек (TypeVar, Generic). Вбудовані структури даних мови Python. Словники. Базові операції над словниками. Приклади задач з використанням словників).	Лекція, самостійна робота	[4, 5, 8, 9, 12-14, 16]	2, 3	1 тиждень
	Тема 2. Робота з різними типами даних (Числові задачі. Особливості числових типів. Методи налагодження і тестування. Задачі опрацювання текстових даних і даних спеціальних форматів. Системи кодування літер. Використання бібліотек TypeVar, Generic.). <i>Індивідуальне завдання №1.</i>	Лабораторна, самостійна робота	[4, 5, 8, 9, 12-14, 16]	2, 3	Під час заняття 3 тижні
3,4	Тема 3. Регулярні вирази (Основні принципи роботи з регулярними виразами за допомогою бібліотеки re).	Лекція, самостійна робота	[4, 5, 8, 9, 12-14, 16]	4, 4	2 тижні
	Тема 3. Регулярні вирази (Основні принципи роботи з регулярними виразами за допомогою бібліотеки re. Пошук числових, буквенних та комбінованих частин тексту у файлах).	Лабораторна, самостійна робота	[8, 9, 12-14, 16]	4, 3	Під час заняття

5	Тема 4. Робота з файлами та каталогами (Режими відкриття файлу. Сканери файлів. Модулі та інструменти для роботи з файлами і каталогами. Типові задачі).	Лекція, самостійна робота	[8, 9, 12-14, 16]	2, 4	1 тиждень
	Тема 4. Робота з файлами та каталогами (Задачі пошуку файлів за різними критеріями. Відображення характеристик файлу. Задачі операцій з файлами і каталогами. Інструменти розробника). <i>Приймання індивідуального завдання №1.</i> <i>Індивідуальне завдання №2.</i>	Лабораторна, самостійна робота	[8, 9, 12-14, 16]	2, 3	Під час заняття 2 тижні
6	Тема 5. Робота з API (Приклад проекту з використанням бібліотек requests та plotly).	Лекція, самостійна робота	[3-5, 8-10, 12-16]	2, 7	1 тиждень
	Тема 5. Робота з API (Розробка проекту з побудовою онлайн-карти з використанням бібліотек requests та plotly).	Лабораторна, самостійна робота	[3-5, 8-10, 12-16]	2, 5	Під час заняття
7	Тема 6. Елементи теорії ігор (Поняття матричної гри. Рівновага Неша. Пакет nashru. Приклад програмної реалізації маричної гри Камінь-Ножниці-Папір).	Лекція, самостійна робота	[8-16]	2, 3	1 тиждень
	Тема 6. Елементи теорії ігор (Найкраща стратегія Неда Старка. Теорія ігор у серіалі Гострі картузи). <i>Приймання індивідуального завдання №2.</i> <i>Індивідуальне завдання №3.</i>	Лабораторна, самостійна робота	[8-16]	2, 3	Під час заняття 2 тижні
8	Тема 7. Елементи алгоритмічної теорії графів (Робота з пакетом NetworkX. Модель карате клубу Захарі. Моделювання соціально-економічних процесів використання можливостей бібліотек Numpy та Networkx).	Лекція, самостійна робота	[2, 8- 16]	2, 5	1 тиждень

	Тема 7. Елементи алгоритмічної теорії графів (Бібліотека NetworkX. Епідемічна модель засобами Numpy, Pandas та NetworkX).	Лабораторна, самостійна робота	[2, 8- 16]	2, 3	Під час заняття
9	Тема 8. Стандарти обміну даними у мережі Json та XML. (Текстовий формат обміну даними Json. Модель проєкта на основі Json-даних веб-серверів. Постановка задачі. Отримання json-даних. Дослідження структури json-файла. Приклади задач опрацювання json-даних. Архітектура проєкта).	Лекція, самостійна робота	[3-9, 16]	2, 4	1 тиждень
	Тема 8. Стандарти обміну даними у мережі Json та XML. (Формат даних Json. Сервери і ресурси формату Json. Перетворення json-файлів в python-структури. Задачі опрацювання даних формату Json). <i>Приймання індивідуального завдання №3.</i>	Лабораторна, самостійна робота	[3-9, 16]	2, 3	Під час заняття
10	Тема 8. Стандарти обміну даними у мережі Json та XML. (Текстовий формат обміну даними XML. Модель проєкта на основі XML-даних. Постановка задачі. Отримання XML-даних. Дослідження структури XML-файла. Приклади задач опрацювання XML-даних. Архітектура проєкта). <i>Колоквіум</i>	Лекція, самостійна робота	[3-9, 16]	2, 4	1 тиждень Під час заняття
	Тема 8. Стандарти обміну даними у мережі Json та XML. (Планування архітектури проєкта на основі XML-даних серверів. Розробка головного сценарію мовою Python). <i>Індивідуальне завдання №4.</i>	Лабораторна, самостійна робота	[3-9, 16]	2, 3	Під час заняття 3 тижні
11	Тема 9. Використання баз даних. (Поняття про бази даних і їх види. Робота з базами даних в python. Бібліотеки для роботи з реляційними (sqlite3) та нереляційними базами даних (pymongo)).	Лекція, самостійна робота	[2, 6, 7, 10, 12, 13]	2, 4	1 тиждень

	Тема 9. Використання баз даних. (Поняття про бази даних і їх види. Робота з базами даних в python. Бібліотеки для роботи з реляційними (sqlite3) та нереляційними базами даних (pymongo)).	Лабораторна, самостійна робота	[2, 6, 7, 10, 12, 13]	2, 3	Під час заняття
12	Тема 10. Наукові обчислення за допомогою python (numpy, scipy, Scikit-learn, pandas, etc.)	Лекція, самостійна робота	[1, 6, 7, 12, 13]	2, 4	1 тиждень
	Тема 10. Наукові обчислення за допомогою python (numpy, scipy, Scikit-learn, pandas, etc.)	Лабораторна, самостійна робота	[1, 6, 11, 12]	2, 3	Під час заняття
13	Тема 11. Аналіз даних з пропусками (Механізми породження пропусків. Методи заповнення пропусків. Заповнення середніми. Заповнення з підбором. Заповнення за регресією).	Лекція, самостійна робота	[1, 6, 7, 12, 13]	2, 4	1 тиждень
	Тема 11. Аналіз даних з пропусками (Механізми породження пропусків. Методи заповнення пропусків. Заповнення середніми. Заповнення з підбором. Заповнення за регресією). <i>Приймання індивідуального завдання №4.</i>	Лабораторна, самостійна робота	[1, 6, 11, 12]	2, 3	Під час заняття
14	Тема 12. Тестування і налагодження задач проєкта <i>Комплексне контрольне опитування</i>	Лекція, самостійна робота	[2, 12-16]	2, 4	1 тиждень Під час заняття
	Тема 12. Тестування і налагодження задач проєкта Програмування, тестування і налагодження задач проєкта (C#, Python тощо). Комплексна перевірка цілого проєкта. Документування для користувача.	Лабораторна, самостійна робота	[2, 12-16]	2, 4	Під час заняття